

PENENTUAN MODEL RETURN HARGA SAHAM DENGAN MULTI LAYER FEED FORWARD NEURAL NETWORK MENGGUNAKAN ALGORITMA RESILENT BACKPROPAGATION

(Studi Kasus : Harga Penutupan Saham Unilever Indonesia Tbk. Periode September
2007 – Maret 2015)

Riza Adi Priantoro¹, Dwi Ispriyanti², Moch. Abdul Mukid³

¹Mahasiswa Jurusan Statistika FSM Universitas Diponegoro

^{2,3}Staff Pengajar Jurusan Statistika FSM Universitas Diponegoro

riza49772@gmail.com, ispriyanti.dwi@gmail.com, mamukid@yahoo.com

ABSTRACT

Determination of a return of stock price model is often associated with a process of forecasting for future periods. A method that can be used is neural network. The use of neural network in the field of forecasting can be a good solution, but the problem is how to determine the network architecture and the selection of appropriate training methods. One possible option is to use resilient back propagation algorithm. Resilient back propagation algorithm is a supervised learning algorithm to change the weights of the layers. This algorithm uses the error in the backward direction (back propagation), but previously performed advanced stage (feed forward) to get the error. This algorithm can be used as a learning method in training model of a multi-layer feed forward neural network. From the results of the training and testing on the share return of stock price PT. Unilever Indonesia Tbk. data obtained MSE value of 0.0329. This model is good to use because it provides a fairly accurate prediction of the results shown by the proximity of the target with the output.

Keywords : return, neural network, back propagation, feed forward, back propagation algorithm, weight, forecasting.

1. PENDAHULUAN

Perkembangan dunia investasi tidak saja ditunjukkan oleh semakin meningkatnya jumlah uang yang diinvestasikan ataupun oleh semakin banyaknya jumlah investor yang berinvestasi. Akan tetapi perkembangan tersebut juga ditunjukkan oleh semakin banyaknya alternatif-alternatif instrumen investasi yang bisa dijadikan pilihan investor dalam berinvestasi. Salah satunya yaitu pasar saham. Saham adalah surat berharga yang menunjukkan bagian kepemilikan atas suatu perusahaan. Sebelum melakukan investasi perlu menganalisis terlebih dahulu saham yang akan dibeli, apakah akan menghasilkan keuntungan atau sebaliknya akan menghasilkan kerugian bagi investor. Untuk mempermudah menganalisis keuntungan atau kerugian yang akan diperoleh investor dapat dilakukan dengan menganalisis return harga saham. Return merupakan hasil yang akan diperoleh dari investasi yang dilakukan oleh investor, untuk memudahkan menganalisis besarnya return yang akan diperoleh investor dapat dilakukan pemodelan terlebih dahulu. Salah satu metode yang dapat digunakan untuk memodelkan nilai return yaitu model *multi layer feed forward neural network*. Model ini terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output dengan algoritma pelatihan yang digunakan salah satunya

menggunakan algoritma *resilient backpropagation* yang mampu mempercepat fungsi kinerja jaringan serta mempercepat pencapaian bobot yang konvergen.

Tujuan dari penelitian ini untuk menentukan model *multi layer feed forward neural network* dalam menghitung nilai return harga saham dan memprediksi nilai return harga saham pada periode mendatang dengan model *multi layer feed forward neural network* yang didapat.

2. TINJAUAN PUSTAKA

2.1 Return Saham

Return atau pengembalian adalah keuntungan yang diperoleh perusahaan, individu, dan investor dari hasil kebijakan investasi yang dilakukan. Nilai return dapat dicari dengan perbandingan nilai harga saham pada waktu ke- t (h_t) dan harga saham pada waktu $t-1$ (h_{t-1}). Nilai return akan bernilai positif jika terjadi kenaikan harga saham, dan bernilai negatif jika terjadi penurunan harga saham, sehingga fluktuasi harga saham dapat terlihat lebih jelas jika menggunakan data return. Return saham dapat dihitung secara harian, mingguan, bulanan, dan juga tahunan. Menurut Ghozali (2007) untuk dapat mengetahui besar nilai return harga saham (x_t) dapat didefinisikan dengan :

$$x_t = \frac{(h_t - h_{t-1})}{h_{t-1}}$$

2.2 Aspek-aspek neural network

Jaringan *neural network* memiliki tiga elemen dasar dalam melakukan pemrosesan informasi, yaitu himpunan penghubung, fungsi penjumlah, dan fungsi aktivasi.

1. Himpunan penghubung merupakan himpunan unit-unit yang dihubungkan oleh satu jalur koneksi dimana jalur-jalur tersebut memiliki bobot yang berbeda.
2. Fungsi penjumlah merupakan suatu unit yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobot masing-masing, yang dirumuskan sebagai berikut:

$$y_{in\ j} = \sum_i^n w_{ji} x_i$$

3. Fungsi aktivasi dapat menghasilkan informasi yang akan dikirim ke neuron lain dengan bobot kedatangan tertentu, selain itu fungsi aktivasi dapat digunakan untuk mengubah bobot.

2.3 Fungsi Aktivasi

Fungsi aktivasi digunakan untuk mengaktifkan fungsi penjumlah menjadi output jaringan. Menurut Kusumadewi (2004) apabila hasil fungsi penjumlahan melewati suatu nilai ambang maka neuron tersebut akan diaktifkan, namun jika tidak melewati nilai ambang maka neuron tersebut tidak perlu diaktifkan. Ada beberapa fungsi aktivasi yang dapat digunakan pada jaringan syaraf tiruan yang menggunakan propagasi balik, yaitu:

1. Fungsi Linear (*Identitas*)

Fungsi linear memberikan output yang sama dengan input data yang dimasukan. Maka fungsi linear dapat dirumuskan sebagai berikut:

$$y = f(x) = x.$$

Dengan turunan fungsi linear sebagai berikut :

$$f'(x) = 1$$

2. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, yang membedakannya adalah output yang dihasilkan oleh fungsi sigmoid bipolar berada pada range -1 sampai dengan 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut :

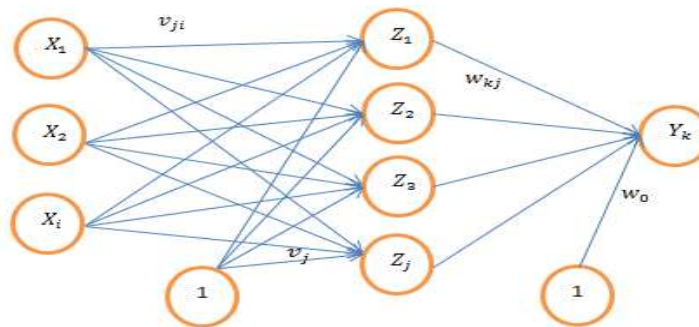
$$y = f(x) = \frac{1 - e^{-\sigma x}}{1 + e^{-\sigma x}}.$$

Dengan turunan fungsi sigmoid bipolar sebagai berikut :

$$f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)]$$

2.4. Arsitektur Neural Network

Neural Network tersusun dari lapisan–lapisan (*layers*) yang memiliki pola keterhubungan baik dalam satu lapis atau antar lapis. Dimana didalam lapisan–lapisan yang membentuk sebuah jaringan terdapat elemen pemrosesan sederhana yang disebut dengan neuron. Setiap neuron dihubungkan dengan neuron yang ada pada lapisan sebelum dan sesudahnya (kecuali lapisan input dan output) melalui pola hubungan yang disebut arsitektur jaringan. Arsitektur jaringan dapat ditunjukkan pada Gambar berikut :



Jaringan Multi Lapis

2.5. Algoritma Reilent Backpropagation

Algoritma *resilent backpropagation* adalah algoritma pembelajaran terawasi yang digunakan untuk mengubah bobot–bobot yang terhubung dengan neuron–neuron yang ada pada lapisan tersembunyi. Algoritma ini menggunakan tanda turunan parsial untuk mengubah nilai bobot dalam arah mundur, namun sebelumnya membutuhkan nilai eror dengan menggunakan tahap maju terlebih dahulu. Pada awal itersai besarnya perubahan bobot (Δv_{ji} , Δw_{kj} , Δv_j , dan Δw_k) diinisialisasikan dengan parameter $\Delta 0$. Besarnya perubahan bobot tidak boleh melebihi batas maksimum yang ditetapkan pada parameter Δ_{max} , apabila perubahan bobot melebihi Δ_{max} maka perubahan bobot sama dengan Δ_{max} .

Pada proses awal setiap iterasi, simpan (ϕ_{kj} , ϕ_{ji} , β_k , dan β_j) sebagai eror lama, kemudian hitung $[(\phi_{kj} * \phi_{kj} \text{ lama}), (\beta_k * \beta_k \text{ lama}), (\phi_{ji} * \phi_{ji} \text{ lama}), (\text{dan } \beta_j * \beta_j \text{ lama})]$ kemudian hitung koreksi bobot antar lapisan :

memperbaiki bobot pada lapisan tersembunyi ke lapisan output (w_{kj}) :

$$\Delta w_{kj} = \begin{cases} \text{delt_inc} * \Delta w_{kj} (\text{old}); & \varphi_{kj} * \varphi_{kj} \text{ lama} > 0 \\ \text{delt_dec} * \Delta w_{kj} (\text{old}); & \varphi_{kj} * \varphi_{kj} \text{ lama} < 0 \\ \Delta w_{kj} (\text{old}); & \varphi_{kj} * \varphi_{kj} \text{ lama} = 0 \end{cases}$$

$$\Delta w_{kj} = \min(\Delta w_{kj}, \text{deltamax})$$

$$\Delta w_{kj} = \begin{cases} -\Delta w_{kj}; & \varphi_{kj} > 0 \\ \Delta w_{kj}; & \varphi_{kj} < 0 \\ 0; & \varphi_{kj} = 0 \end{cases}$$

memperbaiki bobot pada unit bias ke unit output (w_k) :

$$\Delta w_k = \begin{cases} \text{delt_inc} * \Delta w_k (\text{old}); & \beta_k * \beta_k \text{ lama} > 0 \\ \text{delt_dec} * \Delta w_k (\text{old}); & \beta_k * \beta_k \text{ lama} < 0 \\ \Delta w_k (\text{old}); & \beta_k * \beta_k \text{ lama} = 0 \end{cases}$$

$$\Delta w_k = \begin{cases} \min(\Delta w_k, \text{deltmax}); & \beta_k * \beta_k \text{ lama} \geq 0 \\ \max(\Delta w_k, \text{deltmin}); & \beta_k * \beta_k \text{ lama} < 0 \end{cases}$$

$$\Delta w_k = \begin{cases} -\Delta w_k; & \beta_k > 0 \\ \Delta w_k; & \beta_k < 0 \\ 0; & \beta_k = 0 \end{cases}$$

memperbaiki bobot pada lapisan input ke lapisan tersembunyi (v_{ji}) :

$$\Delta v_{ji} = \begin{cases} \text{delt_inc} * \Delta v_{ji} (\text{old}); & \varphi_{ji} * \varphi_{ji} \text{ lama} > 0 \\ \text{delt_dec} * \Delta v_{ji} (\text{old}); & \varphi_{ji} * \varphi_{ji} \text{ lama} < 0 \\ \Delta v_{ji} (\text{old}); & \varphi_{ji} * \varphi_{ji} \text{ lama} = 0 \end{cases}$$

$$\Delta v_{ji} = \min(\Delta v_{ji}, \text{deltamax})$$

$$\Delta v_{ji} = \begin{cases} -\Delta v_{ji}; & \varphi_{ji} > 0 \\ \Delta v_{ji}; & \varphi_{ji} < 0 \\ 0; & \varphi_{ji} = 0 \end{cases}$$

memperbaiki bobot pada unit bias ke lapisan tersembunyi (v_j) :

$$\Delta v_j = \begin{cases} \text{delt_inc} * \Delta v_j(\text{old}); & \beta_j * \beta_j \text{ lama} > 0 \\ \text{delt_dec} * \Delta v_j(\text{old}); & \beta_j * \beta_j \text{ lama} < 0 \\ \Delta v_j(\text{old}); & \beta_j * \beta_j \text{ lama} = 0 \end{cases}$$

$$\Delta v_j = \begin{cases} \min(\Delta v_j, \text{deltmax}); & \beta_j * \beta_j \text{ lama} \geq 0 \\ \max(\Delta v_j, \text{deltmin}); & \beta_j * \beta_j \text{ lama} < 0 \end{cases}$$

$$\Delta v_j = \begin{cases} -\Delta v_j; & \beta_j > 0 \\ \Delta v_j; & \beta_j < 0 \\ 0; & \beta_j = 0 \end{cases}$$

2.6. Model Multi Layer Feed Forward Neural Network

Model *multi layer feed forward neural network* yang terdiri dari satu lapis input dengan n neuron, satu lapis tersembunyi dengan p unit neuron, dan satu lapis output dengan q neuron. Model yang didapat dapat dinotasikan dengan (n-p-q). Dapat dituliskan dalam persamaan berikut :

$$Y_k = \psi_0 \left\{ w_k + \sum_{j=1}^p w_{kj} \psi_n \left(v_j + \sum_{i=1}^n v_{ji} X_i \right) \right\}$$

3. METODE PENELITIAN

Pada penelitian ini data yang digunakan adalah data sekunder dengan jenis data kuantitatif, yaitu data harga penutupan saham Unilever Indonesia Tbk. data diambil pada tanggal 14 Maret 2015 periode bulan Oktober 2007–Maret 2015 yang diperoleh dari internet (<http://finance.yahoo.com>) yang kemudian dicari nilai returnnya dengan teknik pengolahan data yang dilakukan adalah sebagai berikut :

1. Menentukan nilai return harga saham.
2. Membuat plot PACF (*Partial Auto Corelation Function*) untuk menentukan variabel yang akan dijadikan komponen input.
3. Membagi data menjadi dua bagian, yaitu data pelatihan dan data uji .
4. Menentukan jumlah unit neuron pada lapisan tersembunyi.
5. Inisialisasi atau pemilihan bobot awal.
6. Proses pelatihan (*training*)
 - a. Menjumlahkan input yang sudah dikalikan dengan bobotnya (fungsi penjumlah) pada lapisan tersembunyi.
 - b. Mengaktivasi fungsi penjumlah dan meneruskan ke lapisan output
 - c. Menjumlahkan input yang sudah dikalikan dengan bobotnya (fungsi penjumlah) pada lapisan output
 - d. Mengaktivasi fungsi penjumlah
 - e. Perhitungan eror lapisan output ke lapisan tersembunyi
 - f. Perubahan bobot pada lapisan tersembunyi

- g. Perhitungan eror lapisan tersembunyi ke lapisan input
- h. Perubahan bobot pada lapisan input
- 7. Simpan perubahan bobot
- 8. Penentuan model
- 9. Proses pengujian (*testing*)
 - a. Menjumlahkan input yang sudah dikalikan dengan bobotnya (fungsi penjumlah) pada lapisan tersembunyi
 - b. Mengaktivasi fungsi penjumlah dan meneruskan ke lapisan output
 - c. Menjumlahkan input yang sudah dikalikan dengan bobotnya (fungsi penjumlah) pada lapisan output
 - d. Mengaktivasi fungsi penjumlah
- 10. Peramalan return harga saham untuk periode mendatang.

4. HASIL DAN PEMBAHASAN

- a. Menentukan input jaringan
 Penentuan input jaringan dilakukan dengan menggunakan plot PACF hal ini dikarenakan hubungan antara data yang akan dicari dipengaruhi oleh data pada periode sebelumnya. Input jaringan yang digunakan adalah lag-lag yang signifikan pada plot PACF. Lag yang signifikan adalah lag yang melewati garis signifikansi (garis putus-putus berwarna merah).
- b. Pembagian data
 Setelah diketahui variabel input yang digunakan dari plot PACF, data dibagi menjadi 2 yaitu data *training* dan data *testing* dengan komposisi data yang digunakan pada tugas akhir ini 75% untuk data *training* sebanyak 55 data dan 25% untuk data *testing* sebanyak 19 data.
- c. Normalisasi data
 Sebelum dilakukan proses *training*, data perlu dinormalisasi terlebih dahulu. Proses normalisasi dapat dilakukan dengan bantuan mean dan standardeviasi, dengan rumus :

$$norm = \frac{x_t - \bar{x}}{s}$$

- d. Menentukan jumlah unit neuron pada lapis tersembunyi
 Penentuan jumlah unit neuron pada lapis tersembunyi dilakukan dengan mencoba-coba mulai dari 3 neuron sampai 11 neuron menggunakan algoritma *resilient backpropagation*. Indikator yang digunakan untuk menentukan jumlah unit neuron yang optimum pada model menggunakan MSE. menunjukkan nilai MSE terendah sebesar 0,0396 dengan jumlah neuron pada lapisan tersembunyi yang digunakan sebanyak 10 neuro. Maka dari itu jaringan yang akan dibentuk menggunakan 10 unit neuron pada lapisan tersembunyi.
- e. Membangun jaringan *multi layer feed forward neural network*
 Setelah didapat jumlah unit neuron yang optimum untuk lapisan tersembunyi dengan memilih MSE terendah, dibangun jaringan *feed forward* menggunakan perintah *newff* pada MATLAB, yaitu

$$net = newff(minmax(pn), [10 \ 1], \{ 'tansig' \ 'purelin' \}, 'trainrp')$$
- f. Model *multi layer feed forward neural network*

$$Y_t = w_0 + \sum_{j=1}^{10} w_{kj} \cdot \frac{1 - e^{-(v_j + \sum_{i=1}^3 v_{ji} X_i)}}{1 + e^{-(v_j + \sum_{i=1}^3 v_{ji} X_i)}}$$

g. Hasil Peramalan

Model return harga saham dengan *multi layer feed forward neural network* menggunakan algoritma *resilient backpropagation* diperoleh hasil prediksi return harga saham seperti pada Tabel 3. Diperoleh hasil prediksi besarnya return harga saham untuk bulan Juni mengalami kenaikan, namun pada bulan April, Mei, Juli, Agustus, dan September mengalami penurunan.

Tabel 3. Hasil Prediksi Return Saham

Tahun	Bulan	Prediksi
2015	April	- 0,0409
	Mei	- 0,0690
	Juni	0,1245
	Juli	- 0,1201
	Agustus	- 0,1871
	September	- 0,2008

5. KESIMPULAN

- Didapat model *multi layer feed forward neural network* dengan algoritma *resilient backpropagation* yang akan digunakan untuk meramalkan return harga saham yang terdiri dari 3 unit neuron pada lapisan input, 10 unit neuron pada lapisan tersembunyi, dan 1 unit neuron pada lapisan output dengan fungsi aktivasi yang digunakan pada lapisan input menuju lapisan tersembunyi menggunakan fungsi aktivasi sigmoid bipolar dan fungsi aktivasi yang digunakan pada lapisan tersembunyi ke lapisan output menggunakan fungsi aktivasi linear (identitas), dengan pelatihan menggunakan algoritma *resilient backpropagation* dapat digunakan untuk meramalkan return harga saham, dengan model yang didapat sebagai berikut :

$$Y_t = w_0 + \sum_{j=1}^{10} w_{kj} \cdot \frac{1 - e^{-(v_j + \sum_{i=1}^3 v_{ji} X_i)}}{1 + e^{-(v_j + \sum_{i=1}^3 v_{ji} X_i)}}$$

- Dengan menggunakan model yang didapat, diperoleh hasil prediksi besarnya return harga saham untuk bulan Juni mengalami kenaikan, namun pada bulan April, Mei, Juli, Agustus, dan September mengalami penurunan.

6. DAFTAR PUSTAKA

- [1] Ghozali, I., 2007, *Manajemen Risiko Perbankan*, BP UNDIP, Semarang
- [2] Kusumadewi, S., 2004, *Membangun Jaringan Syaraf Tiruan menggunakan MATLAB & Excel Linnk*, Graha Ilmu, Yogyakarta
- [3] Warsito, B., 2009, *Kapita Slekta Statistika Neural Network*, BP UNDIP, Semarang